JACKSON LAWRENCE
00000070612

LOUIS GABRIEL HERNANDES
00000070250

FUZZY TOPSIS

OSCAR JIRO HARLISON
00000072786

IGNATIUS STEVEN
00000070642

DEFINITION

- **Fuzzy TOPSIS** adalah sebuah perpanjangan dari teknik TOPSIS tradisional yang menginkorporasi fuzzy logic untuk mengatasi ketidakpastian dan ambiguitas pada MCDM

- Fuzzy TOPSIS bekerja dengan **memeringkatkan alternatif berdasarkan jarak geometris** terdekat dari solusi ideal **fuzzy** positif dan terjauh dari negatif (Izdihar et al., 2023)

- Dikembangkan untuk mengatasi kondisi di mana data presisi tidak mencukupi untuk memodelkan situasi dunia nyata, sehingga fuzzy set digunakan untuk merepresentasikan variabel linguistik untuk menggantikan numerik (Abdelhafeez et al., 2024)

WHY USE FTOPSIS?

- **Mengatasi ketidakpastian dan ambiguitas dengan lebih baik** dibanding pasangan tradisionalnya karena menggunakan angka fuzzy untuk merepresentasikan data yang kurang jelas (B. et al., 2024)
- **Sudah berhasil diaplikasikan dalam banyak kasus dunia nyata** karena cocok pada situasi di dunia nyata yang tidak selalu memiliki data lengkap dan jelas (Abdelhafeez et al., 2024)

HOWEVER...

- **Kompleksitas meningkat dibanding TOPSIS tradisional** karena menggunakan angka fuzzy yang dapat memperumit proses pembuatan keputusan (Yatsalo et al., 2020)
- **Hasil dapat lebih sulit untuk diinterpretasi** karena kurang intuitif untuk para pembuat keputusan (Madi et al., 2016)

RECENT APPLICATIONS

- Manajemen sumber daya manusia (Kodukulla et al., 2024)
- Evaluasi metode pengolahan air limbah (B. et al., 2024)
- Penentuan lokasi kampus (Sahoo et al., 2024)
- Evaluasi performa bandara (Dimitriou et al., 2024)

HOW IT WORKS

- Awalnya, kriteria didefinisikan dan diberikan bobot menggunakan skala fuzzy
- Decision matrix dibentuk berdasarkan alternatif dan bobot setiap kriteria dengan angka fuzzy dan dinormalisasi
- Fuzzy positive dan negative ideal solutions (FPIS & FNIS) diidentifikasi berdasarkan nilai bobot optimal untuk setiap kriteria
- Untuk setiap alternatif, jarak vertex dari FPIS dan FNIS dihitung, lalu dari jarak-jarak tersebut dihitung nilai preferensi atau kedekatan relatif terhadap solusi ideal
- Matriks outranking dibuat dengan menggunakan concordance dan discordance thresholds
- Alternatif-alternatif diurutkan dari nilai preferensi tertinggi

OVERALL CONCEPT

BERIKAN BOBOT PADA KRITERIA

BENTUK DAN NORMALISASI DECISION MATRIX

BERIKAN BOBOT PADA MATRIKS

TENTUKAN FUZZY POSITIVE (FPIS) DAN NEGATIVE IDEAL SOLUTION (FNIS)

HITUNG JARAK VERTEX SETIAP ALTERNATIF DARI FPIS DAN FNIS

HITUNG NILAI RELATIVE CLOSENESS

OUTPUT URUTAN PREFERENSI ALTERNATIF BERDASARKAN NILAI RELATIVE CLOSENESS

KEY COMPONENTS

Fuzzy Decision Matrix

01

Matriks yang mewakili performa setiap alternatif berdasarkan setiap kriteria dalam angka fuzzy.

Fuzzy Criteria Weights

02

Indikasi kepentingan relatif dari setiap kriteria dengan angka fuzzy.

Fuzzy Ideal Solutions

03

Terdiri dari Fuzzy Positive Ideal Solution (FPIS) sebagai solusi terbaik hipotetis dan Fuzzy Negative Ideal Solution (FNIS) sebagai solusi terburuk hipotetis.

Distance Measures

04

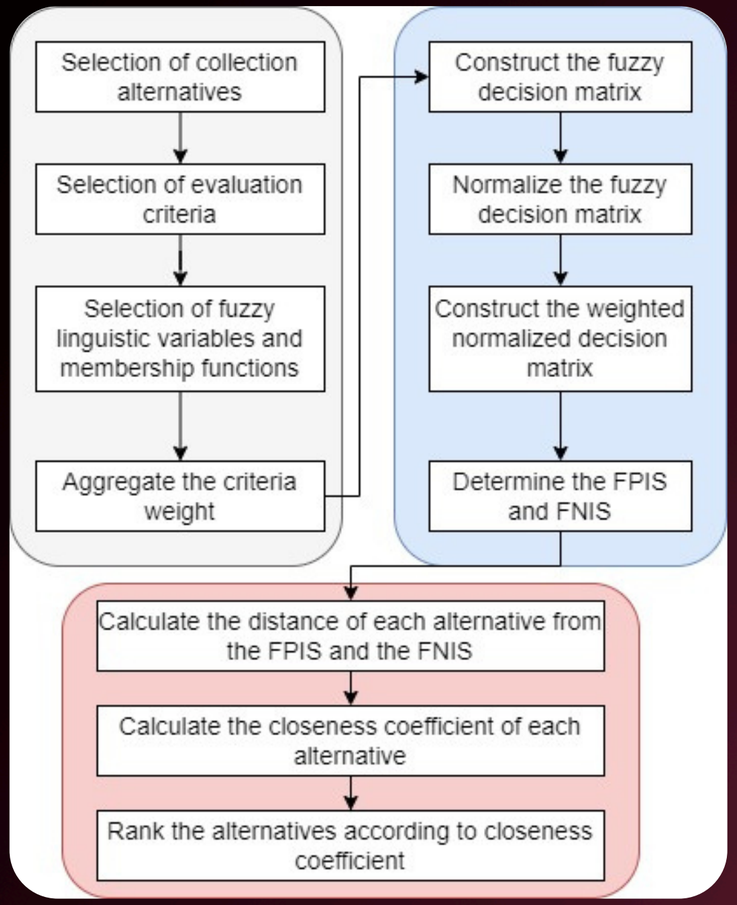
Jarak fuzzy dari setiap alternatif ke PIS dan NIS.

Relative Closeness Score

05

Nilai preferensi atau kedekatan dari solusi ideal untuk pemeringkatan.

FLOWCHART



CODES & ALGORITHM PROCESS

```python
# Fuzzy TOPSIS Method
def fuzzy_topsis(decision_matrix, weights, criteria):
    # Step 1: Normalize the decision matrix
    normalized_matrix = normalize_fuzzy_matrix(decision_matrix, criteria)

    # Step 2: Calculate weighted normalized decision matrix
    weighted_matrix = weighted_normalized_fuzzy_matrix(normalized_matrix, weights)

    # Step 3: Determine the positive and negative ideal solutions
    positive_ideal, negative_ideal = calculate_ideal_solutions(weighted_matrix, criteria)

    # Step 4: Calculate distances to ideal solutions
    pos_distances, neg_distances = calculate_distances(weighted_matrix, positive_ideal, negative_ideal)

    # Step 5: Calculate closeness coefficients
    closeness = calculate_closeness_coefficients(pos_distances, neg_distances)

    return closeness
```

CODES & ALGORITHMS PROCESS

```python
# Normalize the decision matrix based on the type of criteria
def normalize_fuzzy_matrix(matrix, criteria):
    normalized_matrix = []
    for j in range(len(criteria)):
        column = [row[j] for row in matrix]
        if criteria[j] == "max":  # Benefit criteria
            max_val = max(c[2] for c in column)  # Use the upper bound of TFN
            normalized_column = [fuzzy_div(c, max_val) for c in column]
        else:  # Cost criteria
            min_val = min(c[0] for c in column)  # Use the lower bound of TFN
            normalized_column = [fuzzy_div(min_val, c) for c in column]
        normalized_matrix.append(normalized_column)
    return np.array(normalized_matrix).T

# Calculate the weighted normalized decision matrix
def weighted_normalized_fuzzy_matrix(normalized_matrix, weights):
    weighted_matrix = []
    for row in normalized_matrix:
        weighted_row = [fuzzy_multiply(c, w) for c, w in zip(row, weights)]
        weighted_matrix.append(weighted_row)
    return weighted_matrix

# Determine the positive and negative ideal solutions
def calculate_ideal_solutions(weighted_matrix, criteria):
    positive_ideal = []
    negative_ideal = []
    for j in range(len(criteria)):
        column = [row[j] for row in weighted_matrix]
        if criteria[j] == "max":
            positive_ideal.append([max(c[2] for c in column), max(c[1] for c in column), max(c[0] for c in column)])
            negative_ideal.append([min(c[0] for c in column), min(c[1] for c in column), min(c[2] for c in column)])
        else:  # Minimize criteria
            positive_ideal.append([min(c[0] for c in column), min(c[1] for c in column), min(c[2] for c in column)])
            negative_ideal.append([max(c[2] for c in column), max(c[1] for c in column), max(c[0] for c in column)])
    return positive_ideal, negative_ideal
```

```python
# Calculate distances to positive and negative ideal solutions
def calculate_distances(weighted_matrix, positive_ideal, negative_ideal):
    positive_distances = []
    negative_distances = []
    for row in weighted_matrix:
        pos_dist = sum(fuzzy_distance(c, p) for c, p in zip(row, positive_ideal))
        neg_dist = sum(fuzzy_distance(c, n) for c, n in zip(row, negative_ideal))
        positive_distances.append(pos_dist)
        negative_distances.append(neg_dist)
    return positive_distances, negative_distances

# Calculate closeness coefficients
def calculate_closeness_coefficients(pos_distances, neg_distances):
    return [neg / (pos + neg) for pos, neg in zip(pos_distances, neg_distances)]
```

https://github.com/KriezAlf/ExSys/

LINGUISTIC VARIABLES

Linguistic Variables	Fuzzy Value
Very Low (VL)	(0, 0, 0.1)
Low (L)	(0, 0.1, 0.3)
Medium Low (ML)	(0.1, 0.3, 0.5)
Medium (M)	(0.3, 0.5, 0.7)
Medium High (MH)	(0.5, 0.7, 0.9)
High (H)	(0.7, 0.9, 1.0)
Very High (VH)	(0.9, 1.0, 1.0)

***Fuzzy value for importance weight of each criterion.**

Linguistic Variables	Fuzzy Value
Very Poor (VP)	(0, 0, 1)
Poor (P)	(0, 1, 3)
Medium Poor (MP)	(1, 3, 5)
Fair (F)	(3, 5, 7)
Medium Good (MG)	(5, 7, 9)
Good (G)	(7, 9, 10)
Very Good (VG)	(9, 10, 10)

***Fuzzy value for ratings**

***Can be different for each problem or case study as it is a very subjective POV**

CONSTRUCTING COMBINED DECISION MATRIX

$$\tilde{x}_{ij} = (a_{ij}, b_{ij}, c_{ij})$$

$$a_{ij} = \min_{k}\{a_{ij}^{k}\}$$

$$b_{ij} = \frac{1}{K}\sum_{k=1}^{K} b_{ij}^{k}$$

$$c_{ij} = \max_{k}\{c_{ij}^{k}\}$$

WEIGHTED NORMALIZED MATRIX

$$\tilde{v}_{ij} = \tilde{r}_{ij} \times w_j$$

$$A_1 \otimes A_2 = (a_1, b_1, c_1) \otimes (a_2, b_2, c_2)$$
$$= (a_1 * a_2, b_1 * b_2, c_1 * c_2)$$

FUZZY POSITIVE IDEAL SOLUTION (FPIS)

$$A^{*} = (\tilde{v}_1^{*}, \tilde{v}_2^{*}, \ldots\ldots, \tilde{v}_n^{*})$$

$$where \quad \tilde{v}_j^{*} = \max_{i}\{v_{ij3}\}$$

FUZZY NEGATIVE IDEAL SOLUTION (FNIS)

$$A^{-} = (\tilde{v}_1^{-}, \tilde{v}_2^{-}, \ldots\ldots, \tilde{v}_n^{-})$$

$$where \quad \tilde{v}_j^{-} = \min_{i}\{v_{ij1}\}$$

DISTANCE OF EACH ALTERNATIVE TO THE FPIS & TO THE FNIS

$$d(x,y) = \sqrt{\frac{1}{3}\left[(a_1 - a_2)^2 + (b_1 - b_2)^2 + (c_1 - c_2)^2\right]}$$

For FPIS

$$d_i^{*} = \sum_{j=1}^{n} d\left(\tilde{v}_{ij}, \tilde{v}_j^{*}\right)$$

For FNIS

$$d_i^{-} = \sum_{j=1}^{n} d\left(\tilde{v}_{ij}, \tilde{v}_j^{-}\right)$$

NORMALIZED DECISION MATRIX

Linear Scale Transformation method

$$\tilde{r}_{ij} = \left(\frac{a_{ij}}{c_j^{*}}, \frac{b_{ij}}{c_j^{*}}, \frac{c_{ij}}{c_j^{*}}\right) \quad and \quad c_j^{*} = \max_{i}\{c_{ij}\} \quad \textit{Benefit Criteria}$$

$$\tilde{r}_{ij} = \left(\frac{a_j^{-}}{c_{ij}}, \frac{a_j^{-}}{b_{ij}}, \frac{a_j^{-}}{a_{ij}}\right) \quad and \quad a_j^{-} = \min_{i}\{a_{ij}\} \quad \textit{Cost Criteria}$$

***For making the matrix normalized**

CLOSENESS COEFFICIENT FOR EACH ALTERNATIVE

$$CC_i = \frac{d_i^{-}}{d_i^{-} + d_i^{*}}$$

CASE STUDY : MULTI DECISION MAKER ON CHOOSING THE BEST CANDIDATE

Permasalahan pada umumnya telah memberikan matriks keputusan secara langsung dengan nilainya juga. Akan tetapi, studi kasus ini akan dimulai dari pembuatan matriks keputusan hingga akhir.

Decision Maker 1 :

Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Candidate 1	Average	Very High	High
Candidate 2	High	Very High	Average
Candidate 3	Very High	Average	Low
Candidate 4	Low	Average	Low

Decision Maker 2 :

Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Candidate 1	Average	Very High	High
Candidate 2	High	Very High	Average
Candidate 3	Very High	Average	Low
Candidate 4	Low	Average	Very Low

Decision Maker 3 :

Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Candidate 1	Average	High	High
Candidate 2	High	Average	Average
Candidate 3	High	Average	Low
Candidate 4	Very Low	Low	Very Low

Linguistic Variables	Fuzzy Value
Very Low	(1, 1, 3)
Low	(1, 3, 5)
Average	(3, 5, 7)
High	(5, 7, 9)
Very High	(7, 9, 9)

Fuzzy value untuk studi kasus ini.

Ketiga decision maker ini digunakan untuk menilai siapa yang merupakan kandidat terbaik jika dinilai dari skill komunikasi, pengetahuan, dan waktu untuk menyelesaikan masalah.

Substitusi fuzzy value ke dalam tiga decision maker.

Decision Maker 1 : $a^1_{11}, b^1_{11}, c^1_{11}$

Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Candidate 1	3, 5, 7	7, 9, 9	5, 7, 9
Candidate 2	5, 7, 9	7, 9, 9	3, 5, 7
Candidate 3	7, 9, 9	3, 5, 7	1, 3, 5
Candidate 4	1, 3, 5	3, 5, 7	1, 1, 3

Decision Maker 2 : $a^2_{11}, b^2_{11}, c^2_{11}$

Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Candidate 1	5, 7, 9	7, 9, 9	5, 7, 9
Candidate 2	5, 7, 9	5, 7, 9	3, 5, 7
Candidate 3	7, 9, 9	3, 5, 7	1, 1, 3
Candidate 4	1, 3, 5	3, 5, 7	1, 1, 3

Decision Maker 3 : $a^3_{11}, b^3_{11}, c^3_{11}$

Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Candidate 1	3, 5, 7	5, 7, 9	5, 7, 9
Candidate 2	5, 7, 9	3, 5, 7	3, 5, 7
Candidate 3	5, 7, 9	3, 5, 7	1, 3, 5
Candidate 4	1, 1, 3	1, 3, 5	1, 1, 3

Combined Decision Maker :

$\tilde{x}_{11} = (a_{11}, b_{11}, c_{11})$

$b_{11} = 1/K * (b^1_{11} + b^2_{11} + b^3_{11})$
$= 1/3 * (5 + 7 + 5)$
$= 1/3 * 17$
$= 5.667$

$a_{11} = \min(a^1_{11}, a^2_{11}, a^3_{11})$
$= \min(3, 5, 3) = 3$

$c_{11} = \max(c^1_{11}, c^2_{11}, c^3_{11})$
$= \max(7, 9, 7) = 9$

$\therefore \tilde{x}_{11} = (a_{11}, b_{11}, c_{11})$
$= (3, 5.667, 9)$

Hasil dari Combined Decision Maker : a_{11}, b_{11}, c_{11}

Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Candidate 1	3, 5.667, 9	5, 8.333, 9	5, 7, 9
Candidate 2	5, 7, 9	3, 7, 9	3, 5, 7
Candidate 3	5, 8.333, 9	3, 5, 7	1, 2.333, 5
Candidate 4	1, 2.333, 5	1, 4.333, 7	1, 1, 3

Biasanya pada fuzzy TOPSIS telah diberitahu langsung linguistic variable dan nilainya seperti di atas, tetapi pada permasalahan ini dibentuk oleh 3 decision maker.

Diketahui Weightnya adalah seperti berikut.

Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Weight	High	Very High	Average
	5, 7, 9	7, 9, 9	3, 5, 7

Lakukan normalisasi matrixnya dengan menggunakan metode linear scale transformation.

Benefit criteria = Communication skills & Knowledge
Cost criteria = Time to Solve Problem

$c_j^* = c_1^* = \max(9,9,9,5) = 9$ (Com. Skills)
$c_j^* = c_2^* = \max(9,9,7,7) = 9$ (Knowledge)
$a_j^- = a_3^- = \min(5,3,1,1) = 1$ (Time.....)

CASE STUDY : MULTI DECISION MAKER ON CHOOSING THE BEST CANDIDATE

Hasil normalized matrix :

Weight	5, 7, 9	7, 9, 9	3, 5, 7
Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Candidate 1	$\frac{3}{9}, \frac{5.667}{9}, \frac{9}{9}$	$\frac{5}{9}, \frac{8.333}{9}, \frac{9}{9}$	$\frac{1}{9}, \frac{1}{7}, \frac{1}{5}$
Candidate 2	$\frac{5}{9}, \frac{7}{9}, \frac{9}{9}$	$\frac{3}{9}, \frac{7}{9}, \frac{9}{9}$	$\frac{1}{7}, \frac{1}{5}, \frac{1}{3}$
Candidate 3	$\frac{5}{9}, \frac{8.333}{9}, \frac{9}{9}$	$\frac{3}{9}, \frac{5}{9}, \frac{7}{9}$	$\frac{1}{5}, \frac{1}{2.333}, \frac{1}{1}$
Candidate 4	$\frac{1}{9}, \frac{2.333}{9}, \frac{9}{9}$	$\frac{3}{9}, \frac{5.667}{9}, \frac{9}{9}$	$\frac{1}{3}, \frac{1}{1}, \frac{1}{1}$

Atau.....

Weight	5, 7, 9	7, 9, 9	3, 5, 7
Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Candidate 1	0.333, 0.63, 1	0.556, 0.926, 1	0.111, 0.143, 0.2
Candidate 2	0.556, 0.778, 1	0.333, 0.778, 1	0.143, 0.2, 0.333
Candidate 3	0.556, 0.926, 1	0.333, 0.556, 0.778	0.2, 0.429, 1
Candidate 4	0.111, 0.259, 0.556	0.111, 0.481, 0.778	0.333, 1, 1

Kemudian hitung matriks yang ternormalisasi dikali dengan weight value masing-masing criteria.

Tiap column dikali dengan masing-masing nilai dilihat dari tiap koma dan disesuaikan per kolom.

Weight	5, 7, 9	7, 9, 9	3, 5, 7
Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Candidate 1	5*0.333, 7*0.63, 9*1	7*0.556, 9*0.926, 9*1	3*0.111, 5*0.143, 7*0.2
Candidate 2	5*0.556, 7*0.778, 9*1	7*0.333, 9*0.778, 9*1	3*0.143, 5*0.2, 7*0.333
Candidate 3	5*0.556, 7*0.926, 9*1	7*0.333, 9*0.556, 9*0.778	3*0.2, 5*0.429, 7*1
Candidate 4	5*0.111, 7*0.259, 9*0.556	7*0.111, 9*0.481, 9*0.778	3*0.333, 5*1, 7*1

Atau.....

Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Candidate 1	1.667, 4.407, 9	3.889, 8.333, 9	0.333, 0.714, 1.4
Candidate 2	2.778, 5.444, 9	2.333, 7, 9	0.429, 1, 2.333
Candidate 3	2.778, 6.481, 9	2.333, 5, 7	0.6, 2.143, 7
Candidate 4	0.556, 1.815, 5	0.778, 4.333, 7	1, 5, 7

Hitung nilai A* dan A^- :

Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Candidate 1	1.667, 4.407, 9	3.889, 8.333, 9	0.333, 0.714, 1.4
Candidate 2	2.778, 5.444, 9	2.333, 7, 9	0.429, 1, 2.333
Candidate 3	2.778, 6.481, 9	2.333, 5, 7	0.6, 2.143, 7
Candidate 4	0.556, 1.815, 5	0.778, 4.333, 7	1, 5, 7
A*	2.778, 6.481, 9	3.889, 8.333, 9	1, 5, 7
A^-	0.556, 1.815, 5	0.778, 4.333, 7	0.333, 0.714, 1.4

CASE STUDY : MULTI DECISION MAKER ON CHOOSING THE BEST CANDIDATE

Dari nilai A* dan A⁻ dapat menghitung nilai jarak antara A* dengan FPIS dan nilai jarak antara A⁻ dengan FNIS.

$$d(1,1) = \sqrt{\{(1/3)*[(1.667-2.778)^2+(4.407-6.841)^2+(9-9)^2]\}} = 1.358 \text{ (Untuk FPIS)}$$

Secara lengkap, jarak A* dengan FPIS adalah :

Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Candidate 1	1.358	0	4.089
Candidate 2	0.599	1.183	3.564
Candidate 3	0	2.417	1.666
Candidate 4	3.773	3.145	0

$$d(2,1) = \sqrt{\{(1/3)*[(2.778-2.778)^2+(5.444-6.841)^2+(9-9)^2]\}} = 0.599 \text{ (Untuk FNIS)}$$

Secara lengkap, jarak A⁻ dengan FNIS adalah :

Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem
Candidate 1	2.826	3.145	0
Candidate 2	3.372	2.124	0.566
Candidate 3	3.773	0.977	3.34
Candidate 4	0	0	4.089

Kemudian hitung jumlah dari setiap jarak dengan criterianya untuk dihitung closeness coefficient dari setiap alternative.

FPIS :

Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem	d_i^*
Candidate 1	1.358	0	4.089	5.448
Candidate 2	0.599	1.183	3.564	5.346
Candidate 3	0	2.417	1.666	4.083
Candidate 4	3.773	3.145	0	6.919

FNIS :

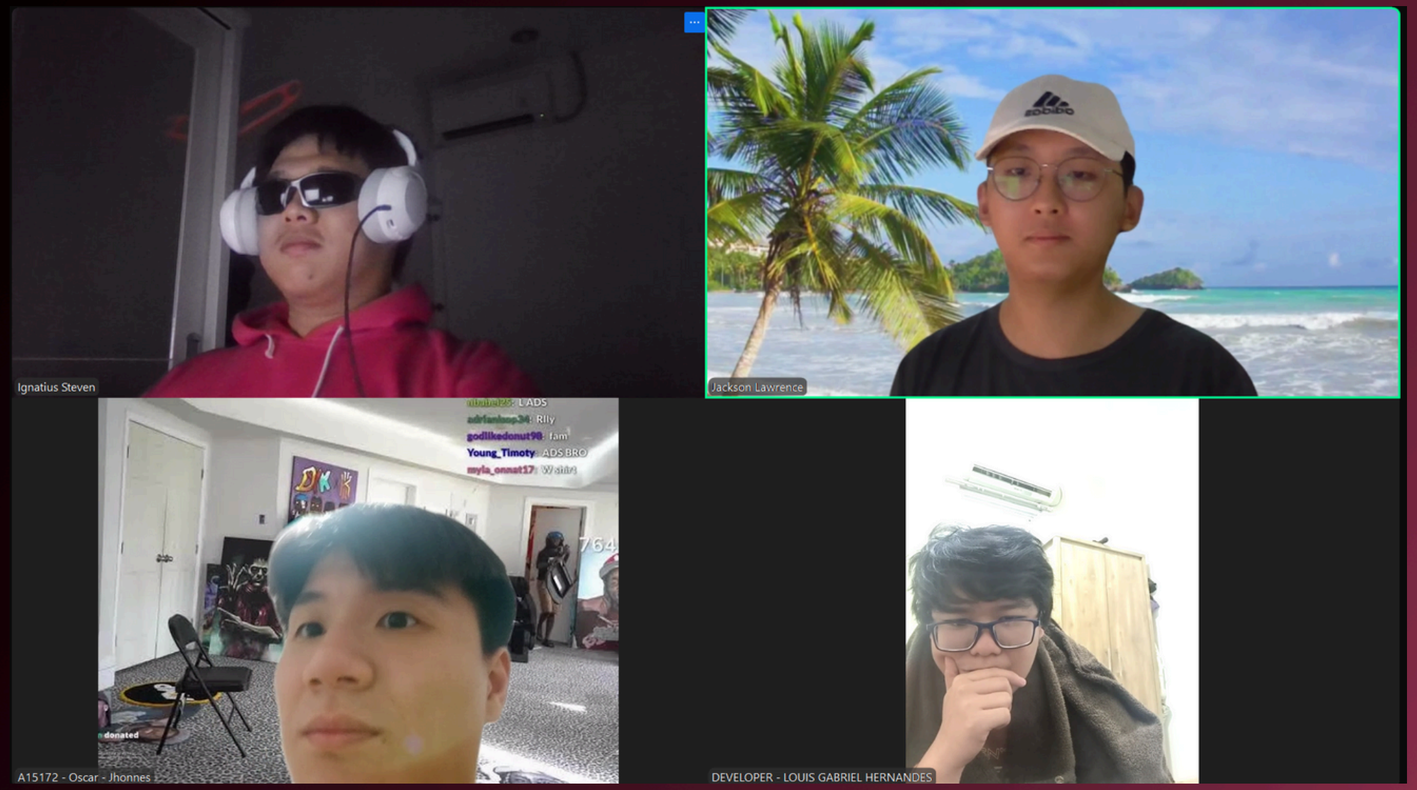
Attribute or Criteria	Communication Skills	Knowledge	Time To Solve Problem	d_i^-
Candidate 1	2.826	3.145	0	5.971
Candidate 2	3.372	2.124	0.566	6.062
Candidate 3	3.773	0.977	3.34	8.091
Candidate 4	0	0	4.089	4.089

Hitung closeness coefficient untuk setiap alternatif untuk mendapat hasil akhir yaitu sebagai berikut.

$CC_1 = 5.971 / (5.448 + 5.971)$
$= 0.522909$
$CC_2 = 6.062 / (6.062 + 5.346)$
$= 0.531406$
$CC_3 = 8.091 / (4.083 + 8.091)$
$= 0.664606$
$CC_4 = 4.089 / (6.919 + 4.089)$
$= 0.371494$

Attribute or Criteria	d_i^*	d_i^-	CC_i	Rank
Candidate 1	5.448	5.971	0.522909	3
Candidate 2	5.346	6.062	0.531406	2
Candidate 3	4.083	8.091	0.664606	1
Candidate 4	6.919	4.089	0.371494	4

∴ Maka, dari perhitungan closeness coefficient dari setiap alternatif, maka didapatkan urutan ranking dengan Candidate 3 adalah kandidat yang terbaik jika dinilai dari performa Communication Skills, Knowledge, dan Time To Solve Problem.



Links:

- https://chatgpt.com/share/673f5e60-36f0-8004-88a7-b9f32a737367

Paper:

- https://www.doi.org/10.18860/jrmm.v2i6.22024
- https://www.doi.org/10.61356/j.iswa.2024.19173
- https://www.doi.org/10.1109/iciteics61368.2024.10625541
- https://www.doi.org/10.1007/S00521-020-05053-9
- https://www.doi.org/10.1109/FUZZ-IEEE.2016.7737950
- https://www.doi.org/10.47191/ijcsrr/v7-i8-28
- https://www.doi.org/10.1016/j.rico.2024.100422
- https://www.doi.org/10.1016/j.rico.2024.100422
- DOI:10.1504/IJOR.2015.072725

Video:

- https://youtu.be/z188EQuWOGU?si=-UhzvFhtgNidxXnS
- https://youtu.be/W3FkLwrXSFM?si=sodm9s8RhaHgld0x
- https://youtu.be/VABG02aiKgE?si=u5ycIFX1EdhYOu5W
- https://youtu.be/ZgdhDYQ3oYw?si=00l4XaC4pDW498_l